

数据库系统原理

陈岭

浙江大学计算机学院

2

关系模型

- 关系和关系模式
- 关系代数
- 扩展的关系代数运算
- 数据库的修改

什么是关系模型？

- ❑ 关系数据库基于关系模型，是一个或多个关系组成的集合
- ❑ 关系通俗来讲就是表（由行和列构成）
- ❑ 关系模型的主要优点是其简单的数据表示，易于表示复杂的查询
- ❑ SQL语言是最广泛使用的语言，用于创建，操纵和查询关系数据库，而关系模型是其基础。

关系示例

Instructor:

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

联系：一些实体之间的关联

关系：是一种数学概念，指的是表
实体集和联系集 \leftrightarrow 真实的世界

关系 - 表，元组 - 行 \leftrightarrow 机器的世界

关系基本结构

- 一般地, 给出集合 D_1, D_2, \dots, D_n , ($D_i = a_{ij} \mid j=1 \dots k$)
 - 关系 r 是: $D_1 \times D_2 \times \dots \times D_n$ 的子集, 即一系列 D_i 域的笛卡尔积
- 因而关系是一组 n 元组 $(a_{1j}, a_{2j}, \dots, a_{nj})$ 的集合, 其中每个 $a_{ij} \in D_i$
- 例如:

{	张清玫教授, 计算机, 李勇 张清玫教授, 计算机, 刘晨 刘逸教授, 信息, 王名
---	--

关系: 导师-专业-学生

笛卡尔积示例

例如: $D_1 = \text{导师集合} = \{\text{张清玫}, \text{刘逸}\}$,
 $D_2 = \text{专业集合} = \{\text{计算机}, \text{信息}\}$,
 $D_3 = \text{研究生集合} = \{\text{李勇}, \text{刘晨}, \text{王名}\}$
 则 $D_1 \times D_2 \times D_3 = \{(\text{张清玫}, \text{计算机}, \text{李勇}),$
 $(\text{张清玫}, \text{计算机}, \text{刘晨}),$
 $(\text{张清玫}, \text{计算机}, \text{王名}),$
 $(\text{张清玫}, \text{信息}, \text{李勇}),$
 $(\text{张清玫}, \text{信息}, \text{刘晨}),$
 $(\text{张清玫}, \text{信息}, \text{王名}),$
 $(\text{刘逸}, \text{计算机}, \text{李勇}),$
 $(\text{刘逸}, \text{计算机}, \text{刘晨}),$
 $(\text{刘逸}, \text{计算机}, \text{王名}),$
 $(\text{刘逸}, \text{信息}, \text{李勇}),$
 $(\text{刘逸}, \text{信息}, \text{刘晨}),$
 $(\text{刘逸}, \text{信息}, \text{王名})\}$

共12个元组

D1	D2	D3
张清玫	计算机	李勇
张清玫	计算机	刘晨
张清玫	计算机	王名
张清玫	信息	李勇
张清玫	信息	刘晨
张清玫	信息	王名
刘逸	计算机	李勇
刘逸	计算机	刘晨
刘逸	计算机	王名
刘逸	信息	李勇
刘逸	信息	刘晨
刘逸	信息	王名

笛卡儿积可用一张二维表表示

导师-专业-学生

张清玫	计算机	李勇
张清玫	计算机	刘晨
刘逸	信息	王名



笛卡尔积示例

dept_name = {Biology, Finance, History, Music}

building = {Watson, Painter, Packard}

budget = {50000, 80000, 90000, 120000}

那么, $r = \{$ (Biology, Watson, 90000),
 (Finance, Painter, 120000),
 (History, Painter, 50000),
 (Music, Packard, 80000)
 $\}$

是 *dept_name* \times *building* \times *budget* 三者间的关系。(共48个元组)

- ❑ 关系的每个属性都有一个名称
- ❑ 域：每个属性的取值集合称为属性的域
- ❑ 属性值必须是原子的，即不可分割(1NF, 第一范式)
 - 多值属性值不是原子的
 - 复合属性值不是原子的
- ❑ 特殊值null是每一个域的成员
- ❑ 空值给数据库访问和更新带来很多困难，因此应尽量避免使用空值
 - 我们先假设不存在空值，在后面的章节中，再讲解空值对不同操作的影响

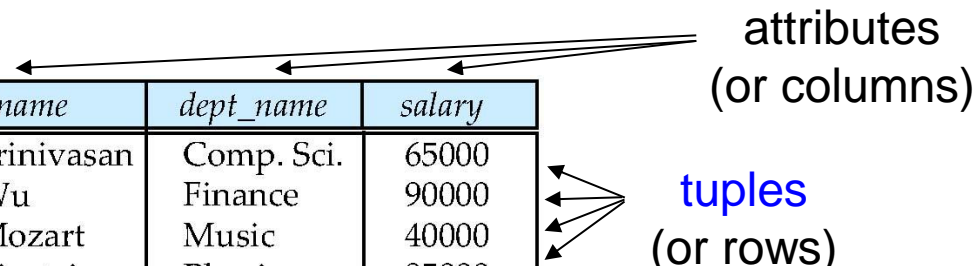
关系的概念

- ❑ 关系涉及两个概念：关系模式和关系实例
- ❑ 关系模式描述关系的结构：
 - 例, *Instructor-schema* = (*ID*: string, *name*: string, *dept_name*: string, *salary*: int)
 - 或 *Instructor-schema* = (*ID*, *name*, *dept_name*, *salary*)
- ❑ 关系实例表示一个关系的特定实例，也就是所包含的一组特定的行
- ❑ 关系、关系模式、关系实例区别：
 - 变量 \leftrightarrow 关系
 - 变量类型 \leftrightarrow 关系模式
 - 变量值 \leftrightarrow 关系实例

- A_1, A_2, \dots, A_n 是属性
- 一般地: $R = (A_1, A_2, \dots, A_n)$ 是一个关系模式
 - 例, *Instructor-schema* = (*ID*, *name*, *dept_name*, *salary*)
- $r(R)$ 是在关系模式 R 上的关系
 - 例, *instructor(Instrutcor-schema)* = *instructor*(*ID*, *name*, *dept_name*, *salary*)

关系实例

- ❑ 关系的当前值（关系实例）由表指定
- ❑ 一个元组 t 代表表中的一行
- ❑ 如果元组变量 t 代表一个元组，那么 $t[name]$ 表示属性 $name$ 的 t 的值



<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

关系的无序性

□ 元组的顺序性是无紧要的（元组能够以任意顺序存储），但一个关系中不能有重复的元组

■ 例，*department*(*dep_name*, *building*, *budget*) 关系的无序显示

<i>dept_name</i>	<i>building</i>	<i>budget</i>
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

- 使 $K \subseteq R$
- 如果K值能够在一个关系中唯一地标志一个元组，则K是R的**超码**
 - 例， $\{instructor-ID, instructor-name\}$ 和 $\{instructor-ID\}$ 都是 *instructor* 的超键
- 如果K是最小超码，则K是**候选码**
 - 例， $\{instructor-ID\}$ 是 *instructor* 的候选码。因为它是一个超码，并且它的任意真子集都不能成为一个超码
- 如果k是一个候选码，并由用户明确定义，则K是一个**主键**。主键通常用下划线标记

外键

□ 假设存在关系r和s: $r(A, B, C)$, $s(B, D)$, 则在关系r上的属性B称作参照s的外码, r也称为外码依赖的参照关系, s叫做外码被参照关系

- 例, 学生(学号, 姓名, 性别, 专业号, 年龄) - 参照关系
专业(专业号, 专业名称) - 被参照关系 (目标关系)
其中属性 专业号 称为关系学生的外码

选修(学号, 课程号, 成绩)

课程(课程号, 课程名, 学分, 先修课号)

- $instructor(ID, name, dept_name, salary)$ - 参照关系
 $department(dept_name, building, budget)$ - 被参照关系

参照关系中外码的值必须在被参照关系中实际存在或为null



大学数据库模式

- ❑ *classroom*(building, room_number, capacity)
- ❑ *department*(dept_name, building, budget)
- ❑ *course*(course_id, title, dept_name, credits)
- ❑ *instructor*(ID, name, dept_name, salary)
- ❑ *section*(course_id, sec_id, semester, year, building, room_number, time_slot_id)
- ❑ *teaches*(ID, course_id, sec_id, semester, year)
- ❑ *student*(ID, name, dept_name, tot_cred)
- ❑ *takes*(ID, course_id, sec_id, semester, year, grade)
- ❑ *advisor*(s_ID, i_ID)
- ❑ *time_slot*(time_slot_id, day, start_time, end_time)
- ❑ *prereq*(course_id, prereq_id)



大学数据库模式图

