

关系型数据库的结构
数据库模式
键
模式图
关系型查询语言
关系代数

Chapter 2: The Relational Model

关系示例

ID	姓名	部门名称	薪水
10101	斯里尼瓦桑	计算机科学	65000
12121	Wu	金融学	90000
15151	莫扎特	音乐	40000
22222	爱因斯坦	物理学	95000
32343	埃尔·赛义德	历史	60000
33456	黄金	物理学	87000
45565	Katz	计算机科学	75000
58583	卡利费里	历史	62000
76543	辛格	金融	80000
76766	克里克	生物学	72000
83821	布兰特	计算机科学	92000
98345	金	电气工程	80000

属性
(或列)
元组
(或行)

基本结构

形式上，给定集合 D_1, D_2, \dots, D_n ，关系 r 是 $D_1 \times D_2 \times \dots \times D_n$ 的子集。因此，关系是 n 元组 (a_1, a_2, \dots, a_n) 的集合，其中每个 $a_i \in D_i$

示例：如果

姓名 = { 吴、莫扎特、戈尔德、辛格…… } /* 所有教师姓名的集合 */
系名 = { 音乐、物理、金融…… } /* 所有系名的集合 */
薪水 = { 40000, 80000, 87000, 90000 … } /* 所有薪水的集合 */
那么 $r = \{ (Wu, Finance, 90000),$
(莫扎特, 音乐系, 40000),
(戈尔德, 物理系, 87000),
(辛格, 财务, 80000) \}

是一个关于
姓名 x 部门名称 x 薪水

关系模式与实例

A_1, A_2, \dots, A_n 是属性

$R = (A_1, A_2, \dots, A_n)$ 是一个关系模式

示例：

教师 = (编号, 姓名, 系名, 薪水)

定义在模式 r 上的关系实例 R 用 $r(R)$ 表示。

关系的当前值由一个表指定

关系 r 的元素 t 称为元组，在表中用一行表示

每个属性的允许值集合称为该属性的域(域)

属性值（通常）要求是原子的(原子的)；即不可分割

特殊值 null (空值) 是每个域的成员。空值会使许多操作的定义变得复杂。

属性

关系是无序的

元组的顺序无关紧要 (元组可以以任意顺序存储)。示例：包含无序元组的教师关系

ID	姓名	系名	薪水
22222	爱因斯坦	物理学	95000
12121	Wu	金融学	90000
32343	埃尔·赛义德	历史学	60000
45565	卡茨	计算机科学	75000
98345	金	电气工程	80000
76766	克里克	生物学	72000
10101	斯里尼瓦桑	计算机科学	65000
58583	卡利费里	历史	62000
83821	布兰特	计算机科学	92000
15151	莫扎特	音乐	40000
33456	黄金	物理学	87000
76543	辛格	金融	80000

示例：

模式：教师 (ID, 姓名, 系名, 工资) 实例：

ID	姓名	系名	薪水
22222	爱因斯坦	物理学	95000
12121	Wu	金融学	90000
32343	赛义德	历史学	60000
45565	卡茨	计算机科学	75000
98345	金	电气工程	80000
76766	克里克	生物学	72000
10101	斯里尼瓦桑	计算机科学	65000
58583	卡利费里	历史	62000
83821	布兰特	计算机科学	92000
15151	莫扎特	音乐	40000
33456	黄金	物理学	87000
76543	辛格	金融	80000

数据库模式

数据库模式——是数据库的逻辑结构。

数据库实例——是数据库中数据在给定时刻的快照。

键

设 $K \subseteq R$

若 K 的值足以识别每个可能关系 $r(R)$ 中的唯一元组，则 K 是 R 的超键（超键）。示例：{ID} 和 {ID、姓名} 都是教师表的超键。若超键 K 是最小的，则 K 是候选键（候选键）。示例：{ID} 是教师表的候选键。候选键之一被选为主键（主键）。

键

从关系 r_1 的属性 A 到关系 r_2 的主键 B 的外键约束表明，在任何数据库实例中， r_1 中每个元组的 A 值必须也是 r_2 中某个元组的 B 值。引用关系：教学

ID	课程编号	被引用关系：教师		
		ID	姓名	部门名称
10101	CS-101	10101	斯里尼瓦桑	计算机科学
12121	FIN-201	12121	Wu	金融
76766	BIO-101	15151	莫扎特	音乐

参照完整性约束要求，在引用关系 r_1 的任何元组的指定属性 A 中出现的值，也必须出现在被引用关系 r_2 中至少一个元组的指定属性 B 中。

引用关系：section					被引用关系：time - slot			
课程编号	课节编号	学期	年	时段ID	时段ID	日	开始时间	
CS-101	1	春天	2022	t1	t1	周二	13:15	
CS-101	2	春天	2022	t1	t1	周五	8:00	
BIO-101	1	夏天	2021	t2	t2	周三	13:15	
					t2	周三	15:40	
						18:30	20:05	

数据库

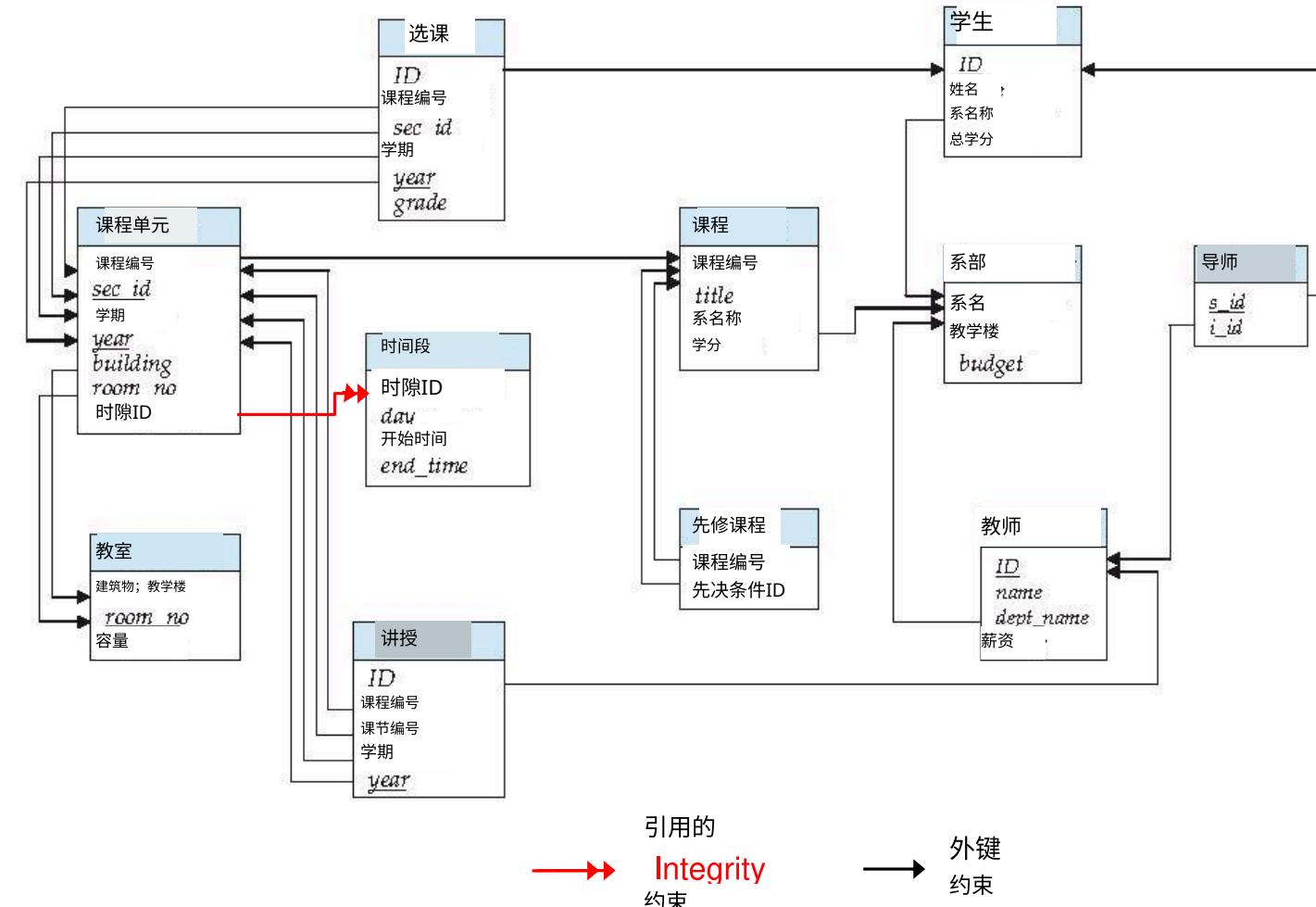
数据库由多个关系组成

企业相关信息被拆分为多个部分：教室（建筑物、房间号、容量）；系（系名、建筑物、预算）；课程（课程编号、课程名称、系名、学分）；教师（ID、姓名、系名、薪水）

课程段（课程编号，课程段编号，学期，年份，教学楼，教室编号，时间段编号）授课（教师编号，课程编号，课程段编号，学期，年份）学生（学生编号，姓名，系名，总学分）选课（学生编号，课程编号，课程段编号，学期，年份，成绩）导师（学生编号，教师编号）时间段（时间段编号，星期，开始时间，结束时间）先修课程（课程编号，先修课程编号）

图2.9 大学数据库的模式。

大学数据库的架构图



关系查询语言

过程式与非过程式（即声明式）

“纯”语言：

关系代数(关系代数)

元组关系演算(元组关系演算)

域关系演算(域关系演算)

上述三种纯语言在计算能力上是等价的

我们将专注于关系代数

不等价于图灵机

由六种基本运算组成

一种过程式语言，由一组操作组成，这些操作以一个或两个关系作为输入，并产生一个新的关系作为结果。六个基本运算符选择： σ 投影； Π 并； \cup 差集； \cdot 笛卡尔积(笛卡尔积)； \times 重命名： ρ

关系代数

Select Operation – Example

关系 r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

$\sigma_{A=B \wedge D>5}(r)$

A	B	C	D
α	α	1	7
β	β	23	10

选择操作会挑选出满足给定谓词的元组
表示法： $\sigma_p(r)$

p 被称为选择谓词

定义为：

$$\sigma_p(r) = \{t \mid t \in r \text{ and } p(t)\}$$

其中 p 是由项组成的命题演算公式

连接方式： κ (与)， v (或)， \neg (非)

每个项为以下之一：

<属性> 操作符 <属性>

<属性> 操作符 <常量>

其中操作符为以下之一：=, ≠, >, ≥, <, ≤

选择示例：

$\sigma_{dept_name='Physics'}(instructor)$

$\sigma_{salary > 90000}(lecturer)$

$\sigma_{dept_name='Physics'} \wedge salary > 90000(lecturer)$

项目操作 - 示例

关系 r:

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

$\Pi_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

A	C
α	1
β	1
β	2

并集操作 - 示例

关系 r, s :

A	B
α	1
α	2
β	1

A	B
α	2
β	3

r 与 s 的并集:

A	B
α	1
α	2
β	1
β	3

投影运算

投影运算是一种一元运算，它返回其参数关系，并去除某些属性。

表示法: $\Pi_{A_1, A_2, \dots, A_k}(r)$

其中 A_1, A_2 是属性名, r 是关系名。

结果定义为通过删除未列出的列而获得的 k 列的关系

由于关系是集合，因此从结果中删除重复行
示例：消除教师表的系名属性

$\Pi_{ID, name, salary}(instructor)$

集合差 - 示例

关系 r, s :

A	B
α	1
α	2
β	1

A	B
α	2
β	3

$r - s$:

A	B
α	1
β	1

r

集合差运算

集合差运算使我们能够找出存在于一个关系中但不存在于另一个关系中的元组。

符号 $r - s$

定义为:

$r - s = \{t \mid t \in r \text{ and } t \notin s\}$

集合差运算必须在兼容的关系之间进行。
 r 和 s 必须具有相同的元数

r 和 s 的属性域必须兼容

示例：查找所有在2009年秋季学期开设，但不在2010年春季学期开设的课程

$\Pi_{course_id}(\sigma_{semester = \mu_{allm}} \wedge year = 2009 \text{ (section)}) -$

$\Pi_{course_id}(\sigma_{semester = "Spring"} \wedge year = 2010 \text{ (section)})$

笛卡尔积运算 - 示例

关系 r, s :

A	B
α	1
β	2

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

$r \times s$:

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

笛卡尔积运算

笛卡尔积运算（用 \times 表示）允许我们将任意两个关系中的信息进行组合。

表示法 $r \times s$

定义为:

$r \times s = \{t q \mid t \in r \text{ and } q \in s\}$

假设 $r(R)$ 和 $s(S)$ 的属性是不相交的。（即 $R \cap S = \emptyset$ ）。

如果 $r(R)$ 和 $s(S)$ 的属性不是不相交的，则必须使用重命名。

运算组合

可以使用多种运算构建表达式

示例: $\sigma_{A=C}(r \times s)$

$r \times s$

	A	B	C	D	E
α	1	α	10	a	
α	1	β	10	a	
α	1	β	20	b	
α	1	γ	10	b	
β	2	α	10	a	
β	2	β	10	a	
β	2	β	20	b	
β	2	γ	10	b	

$\sigma_{A=C}(r \times s)$

	A	B	C	D	E
α	1	α	10	a	
β	2	β	10	a	
β	2	β	20	b	

重命名操作

允许我们为关系代数表达式的结果命名，从而引用这些结果。

允许我们用多个名称来引用一个关系。

示例:

$\rho_x(E)$

以名称 X 返回表达式 E

如果关系代数表达式 E 的元数为 n ，则

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

以名称 X 返回表达式 E 的结果，并将属性重命名为

A_1, A_2, \dots, A_n 。

示例查询

查找物理系所有教师的姓名，以及他们所教授的所有课程的课程编号

查询1

$$\prod_{\text{instructor.name}, \text{course_id}(\sigma_{\text{instructor.ID} = \text{teaches.ID}} (\sigma_{\text{instructor.ID} = \text{teaches.ID}} (\text{instructor.x teaches}))))}$$

查询2

$$\text{教师姓名, 课程编号 } (\sigma_{\text{instructor.ID} = \text{teaches.ID}} (\text{或者系名称} = "物理" (\text{教师}) \times \text{授课}))$$

查询示例

查找物理系所有教师的姓名，以及他们所教授的所有课程的课程编号和课程名称

查询

$$\prod_{\text{教师姓名, 课程编号。课程标题}} (\sigma_{\text{dept_name} = "Physics"} A \text{ 教师编号} = \text{授课教师编号} \text{ 且 } \text{授课课程编号} = \text{课程编号} (\text{教师} \times \text{授课} \times \text{课程}))$$

查询示例

查找该大学中的最高薪资

步骤 1: 找出比其他某些教师薪资低的教师薪资（即非最高薪资）

• 使用一个新名称 d 下的教师表副本

$$\begin{aligned} & \cdot \Pi_{\text{instructor.salary}} (\sigma_{\text{instructor.salary}} < d.\text{salary} (\text{教师} \times \rho_d (\text{教师}))) \\ & \text{步骤 2: 找出最高薪资} - \Pi_{\text{salary}} (\text{教师}) - \backslash (\{\backslash \text{Pi} \\ & \} \text{ }\{\text{text}\{\text{instructor.salary}\}\}\text{ }\leftarrow (\{\backslash \text{sigma} \\ & \} \text{ }\{\text{text}\{\text{instructor.salary}\}\}) \end{aligned}$$

查询示例

查找该大学中的最高薪资

$\Pi_{\text{salary}} (\text{教师})$ -

$$\Pi_{\text{instructor.salary}} (\sigma_{\text{instructor.salary} < d.\text{薪资}} (\text{教师} \times \rho_d (\text{教师})))$$

ID	薪资
10101	70000
12121	80000
15151	90000

教师.ID	教师.薪资	d.ID	d.薪资
10101	70000	10101	70000
10101	70000	12121	80000
10101	70000	15151	90000
12121	80000	10101	70000
12121	80000	12121	80000
12121	80000	15151	90000
15151	90000	10101	70000
15151	90000	12121	80000
15151	90000	15151	90000

形式定义

关系代数中的基本表达式由以下任意一种组成：

数据库中的一个关系

一个常量关系

设 E_1 和 E_2 为关系代数表达式；以下均为关系代数表达式：

$E_1 \cup E_2$

$E_1 - E_2$

$E_1 \times E_2$

$\sigma_p(E_1)$, P 是 E_1 中属性的谓词

$\Pi_S(E_1)$, S 是一个由 E_1 中部分属性组成的列表

$\rho_x(E_1)$, x 是 E_1 结果的新名称

附加操作

我们定义了一些附加操作，这些操作不会给关系代数增添任何能力，但能简化常见查询。

集合交集： $r \cap s$

自然连接： $r \times s$

半连接： α_0

赋值： \leftarrow

外连接： $r \sqsupset s, r \bowtie s, r \supseteq x = s$

除法运算符： $r \div s$

集合交集运算 - 示例

集合交集运算

关系 r, s :

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

$r \cap s$

A	B
α	2

集合交集运算使我们能够找出同时存在于两个输入关系中的元组。

表示法: $r \cap s$

定义为:

$$r \cap s = \{t \mid t \in r \text{ 和 } t \in s\}$$

假设:

r, s 具有相同的元数

r 和 s 的属性兼容

$$\text{注意: } r \cap s = r - (r - s)$$

自然连接 - 示例

关系 r, s :

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ε

s

$r \bowtie s$

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

表示法: $r \times s$

设 r 和 s 分别是模式 R 和 S 上的关系。那么, $r \wedge s$ 是模式 $R \cup S$ 上的一个关系, 其获取方式如下: 考虑来自 r 的每个元组对 t_r 和来自 s 的 t_s , 如果 t_r 和 t_s 在 $R \cap S$ 中的每个属性上具有相同的值, 则将元组 t 添加到结果中, 其中

• t 在 r 上的值与 t_r 相同

• 在 s 上与 t_s 具有相同的值

示例:

$$R = (A, B, C, D)$$

$$S = (E, B, D)$$

$$\text{结果模式} = (A, B, C, D, E)$$

$r \bowtie s =$

$$\prod_{r.A, r.B, r.C, r.D, s.E} (\sigma_{r.B = s.B} \wedge r.D = s.D) (r \times s))$$

自然连接和θ连接

外连接

查找计算机科学系所有教师的姓名, 以及这些教师所教授的所有课程的标题

连接操作的一种扩展, 可避免信息丢失。先计算连接, 然后将一个关系中与另一个关系中的元组不匹配的元组添加到连接结果中。

$$\Pi_{\text{name}, \text{title}} (\sigma_{\text{dept_name} = \text{"Comp. Sci."}} (\text{instructor} \mid \text{it eache} \mid \text{course})) \text{ 自然连接具有结合律}$$

$(\text{instructor} \bowtie \text{teaches}) \bowtie \text{course}$ 等价于
教师 \times (授课 \times 课程)

自然连接具有交换律

教师 \bowtie 授课 等价于
授课 \times 教师

θ 连接操作 $r \bowtie_\theta s$ 定义为

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

使用空值:

空值表示该值未知或不存在

粗略地说, 所有涉及空值的比较在定义上都是假的。

• 我们稍后将研究与空值进行比较的确切含义

外连接 - 示例

外连接 - 示例

教师

授课; 教学

ID	姓名	系名
10101	斯里尼瓦桑	计算机科学
12121	Wu	金融
15151	莫扎特	音乐

ID	课程编号
10101	CS-101
12121	FIN-201
76766	BIO-101

加入

讲师 \bowtie 授课

ID	姓名	系名称	课程编号
10101	斯里尼瓦桑	计算机科学	CS-101
12121	Wu	金融学	FIN-201

左外连接

教师 $\neg X_{\text{teaches}}$

ID	姓名	系名	课程编号
10101	斯里尼瓦桑	计算机科学	CS-101
12121	Wu	金融	FIN-201

教师

授课; 教学

ID	姓名	系名
10101	斯里尼瓦桑	计算机科学
12121	Wu	金融
15151	莫扎特	音乐

右外连接

讲师 $\times \square$ 授课

ID	姓名	系名	课程编号
10101	斯里尼瓦桑	计算机科学	CS-101
12121	Wu	金融学	FIN-201

全外连接

教师 $\exists \times \lceil$ 讲授 \rceil

ID	姓名	系名	课程编号
10101	斯里尼瓦桑	计算机科学	CS-101
12121	Wu	金融	FIN-201

使用连接进行外连接

外连接可以用基本操作来表示

$r \text{ IX } s$ 可以写成

$(r \bowtie s) \cup \left(r - \prod_R (r \bowtie s) \times \{(\text{null}, \dots, \text{null})\} \right)$ $r \times \square s$ 可以写成

$(r \bowtie s)$

$\cup \{(\text{null}, \dots, \text{null})\} \times \left(s - \prod_s (r \bowtie s) \right)$ r 注释 s 可以写成 $(r \times s) \cup (r - \prod_R (r \times s) \times \{ (\text{空}, \dots, \text{空}) \}) \cup \{(\text{null}, \dots, \text{null})\} x \left(s - \prod_s (r \times s) \right)$

半连接操作

表示法: $r \ltimes_{\theta} s$

是 r 的一个子集, 其中每个元组 r_i 在条件 θ 下至少与 s 中的一个元组 s_i 匹配。

$r \ltimes_{\theta} s = \prod_R (r \ltimes_{\theta} s)$

半连接示例:

A	B	C	D
α	1	α	a
1	a	a	
β	2	γ	a
3	a	3	
γ	4	β	b
1	a	γ	
α	1	γ	a
2	b	δ	
δ	2	β	b
3	b	ϵ	

B	D	E
1	a	a
β	2	γ
1	a	γ
α	1	γ
2	b	δ

AD			
α	1	a	a
β	2	γ	a
α	1	γ	a
δ	2	β	b

B	D	E
1	a	a
1	a	γ
2	b	δ

赋值操作

赋值操作 (\leftarrow) 为表达复杂查询提供了一种便捷的方式。

将查询写成由以下内容组成的顺序程序

- 一系列赋值操作
- 其后是一个表达式, 其值作为查询结果显示。

赋值必须始终针对临时关系变量进行。

除法运算 - 示例

关系 r, s :

A	B
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ϵ	6
ϵ	1
β	2

A
α
β

$r \div s$:

A
α

Division Operator

给定关系 $r (R)$ 和 $s (S)$, 使得 $S \subset R, r \div s$ 是最大的关系

$t (R - S)$, 满足
 $t \times s \subseteq r$

例如, 设 $r (ID, course_id) = \prod_{ID, course_id} (\text{takes})$ 和

$s (course_id) = \prod_{course_id} (\text{dept_name} = "Biology") (course)$

那么 $r \div s$ 为我们提供了修完生物学系所有课程的学生

可以将 $r : s$ 写为

```
temp1 ←  $\Pi_{R-S}(r)$ 
temp2 ←  $\Pi_{R-S}((temp1 \times s) - \Pi_{R-S,S}(r))$ 
结果 = 临时变量1 - 临时变量2
```

另一个除法示例

关系 r, s :

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

D	E
a	1
b	1

$r \div s$:

A	B	C
α	a	γ
γ	a	γ

可以在后续表达式中使用变量。

扩展关系代数运算

广义投影
聚合函数

广义投影

通过允许在投影列表中使用算术函数来扩展投影操作。

$\Pi_{F_1, F_2, \dots, F_n}(E)$

E 是任何关系代数表达式

每个 F_1, F_2, \dots, F_n 都是涉及常量和 E 模式中属性的算术表达式。

给定关系 instructor (ID, 姓名, 系名, 工资), 其中工资为年薪, 获取相同信息, 但工资为月薪

$\Pi_{ID, name, dept_name, salary/12}(\text{instructor})$

聚合函数与操作

聚合函数 (Aggregation function, 即“聚合函数”) 接收一组值并返回一个单一值作为结果。avg: 平均值 min: 最小值 max: 最大值 sum: 值的总和 count: 值的数量关系代数中的聚合操作 $G_1, G_2, \dots, G_n \mathcal{G}_{F_1(A_1), F_2(A_2), \dots, F_n(A_n)}(E)$ E 是任意关系代数表达式 G_1, G_2, \dots, G_n 是用于分组的属性列表 (可以为空) 每个 F_i 是一个聚合函数每个 A_i 是一个属性名

聚合操作 - 示例

关系 r:

A	B	C
α	α	7
α	β	7
β	β	3
β	β	10

$\mathcal{G}_{\text{sum}(C)}(r)$

对(C)求和

27

注意: 一些书籍/文章使用 γ 而非 \mathcal{G} (花体 G)

聚合操作 - 示例

查找每个部门的平均工资

部门名称 \mathcal{G} 平均工资 (教师表)

ID	姓名	部门名称	薪水
76766	克里克	生物学	72000
45565	卡茨	计算机科学	75000
10101	斯里尼瓦桑	计算机科学	65000
83821	布兰特	计算机科学	92000
98345	金	电气工程	80000
12121	Wu	金融学	90000
76543	辛格	金融	80000
32343	埃尔·赛义德	历史	60000
58583	卡利费里	历史	62000
15151	莫扎特	音乐	40000
33456	黄金	物理学	87000
22222	爱因斯坦	物理学	95000

部门名称	平均薪资
生物学	72000
计算机科学	77333
电气工程	80000
金融学	85000
历史	61000
音乐	40000
物理学	91000

聚合函数 (续)

聚合结果没有名称

可以使用重命名操作来为其命名

为方便起见, 我们允许在聚合操作中进行重命名

dept_name $\mathcal{G}_{\text{avg}(\text{salary})}$ as avg_sal (instructor)

数据库的修改

可使用以下操作修改数据库内容:

删除

插入

更新

所有这些操作都可以使用赋值运算符来表示

多重集关系代数

纯关系代数会去除所有重复项

例如, 投影之后

多重集关系代数保留重复项, 以符合SQL语义。SQL保留重复项最初是为了提高效率, 但现在已成为一项特性

多重集关系代数定义如下

选择: 如果元组满足选择条件, 则其重复项数量与输入中的相同

投影: 每个输入元组对应一个元组, 即使它是重复项

叉积: 如果在 m 中有 r 个 t1 的副本, 在 n 中有 s 个 t2 的副本, 那么在 $m \times n$ 中有 $r \times s$ 个 t1.t2 的副本

集合运算符

•并集: $m + n$ 副本

•交集: $\min(m, n)$ 副本

•差集: $\min(0, m - n)$ 副本

SQL与关系代数

选择 A_1, A_2, \dots, A_n

从 r_1, r_2, \dots, r_m 中

其中 P

在多重集关系代数中等价于以下表达式

$\Pi_{A_1, \dots, A_n}(\sigma_P(r_1 \times r_2 \times \dots \times r_m))$

选择 A_1, A_2 , 对 A_3 求和

从 r_1, r_2, \dots, r_m 中

其中 P

按 A_1, A_2 分组

等同于多重集关系代数中的以下表达式

$A_1, A_2 \mathcal{G}$ 对(A3)求和 $(\sigma_P(r_1 \times r_2 \times \dots \times r_m))$

SQL与关系代数

更一般地, SELECT子句中的非聚合属性可能是GROUP BY属性的一个子集, 在这种情况下, 等价关系如下:

选择 A_1 , 计算 A_3 的总和

从 r_1, r_2, \dots, r_m 中

其中满足条件 P

按 A_1, A_2 分组

这在多重集关系代数中等价于以下表达式

$\Pi_{A_1, \text{sum } A_3(A_1, A_2)}(\mathcal{G}_{\text{sum}}(A_3) \text{ as } \text{sum } A_3(r_1 \times r_2 \times \dots \times r_m))$

第2章结束