Chapter 6: 形式化关系查询语言

Ch6 形式化关系查询语言

- **The Relational Algebra**：关系代数（重点）
- The Tuple Relational Calculus：元组关系演算
- The Domain Relational Calculus：域关系演算

本章要掌握的内容

- 关键概念：
  - 关系代数的运算特点
    - 运算对象和运算结果都是关系表
  - 关系代数的6个基本运算
    - 每种运算的特点、作用和命令格式
  - 关系代数的附加运算和扩展运算
    - 每种运算的特点、作用和命令格式

- 目的和意义
  - 掌握关系代数的基本运算和运算特点，为设计SQL语言打下基础
- 应用场景
  - 能够根据查询要求写出查询命令

6.1 Relational Algebra

- "纯"语言：
  - 关系代数
  - 元组关系演算
  - 域关系演算
- 上面三种纯语言在计算能力上是等价的

6.1 Relational Algebra

- 关系代数是一种过程化的语言，它由一组运算组成，这些运算用一个或两个关系作为输入，并生成一个新的关系作为它们的结果。
- 6种基本关系代数运算符：
  - 选择：σ
  - 投影：∏
  - 并：∪
  - 集合差：–
  - 笛卡尔积：x
  - 更名：ρ

关系代数运算符

运算符		含义	运算符		含义
集合运算符	$\cup$	并	比较运算符	$>$	大于
	$-$	差		$\geq$	大于等于
	$\cap$	交		$<$	小于
	$\times$	广义笛卡尔积		$\leq$	小于等于
				$=$	等于
				$\neq$	不等于

关系代数运算符

运算符	含义		运算符	含义	
专门的关系运算符	σ π $\bowtie$ $\div$	选择 投影 连接 除	逻辑运算符	$\neg$ $\wedge$ $\vee$	非 与 或

6.1.1 选择运算

- 小组讨论：
  - Select运算有什么特点？
  - 为instructor表设计一个Select运算？

选择运算

- 选择运算选出满足给定谓词的元组
- 符号： $\sigma_p(r)$
- p 叫做选择谓词，r 是进行选择运算的关系
- 例如：查找"Physics"系的教师

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

查询结果见P155，图6-1.

- 在选择谓词中，可以使用比较运算符如：

$$=, \neq, >, \geq, <, \leq$$

- 还可以通过以下连接词将几个谓词组合成一个更长的谓词:

$$\wedge \text{（与）}, \vee \text{（或）}, \neg \text{（非）}$$

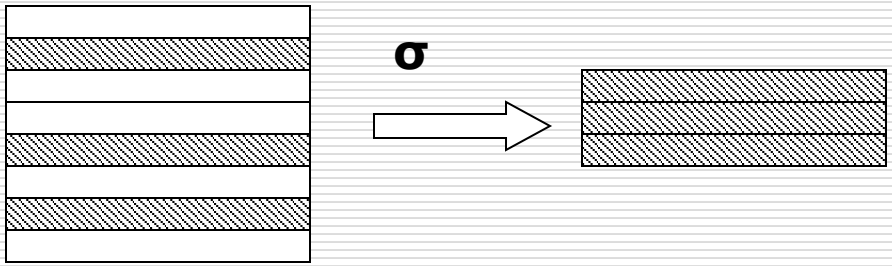
- 例如：查找"Physics"系年薪大于90,000美元的教师

$$\sigma_{\text{dept\_name="Physics"} \wedge \text{ salary>90,000}}(\text{instructor})$$

- 选择谓词可以包含两个属性之间的比较
- 例如：查找系名称与教学楼同名的所有系

$$\sigma_{\text{dept\_name=building}}(\text{department})$$

- 选择运算是从行的角度进行运算



Select Operation – Example

- Relation 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

6.1.2 Project Operation 投影运算

- 小组讨论：
  - Project运算有什么特点？
  - 为instructor表设计一个Project运算？

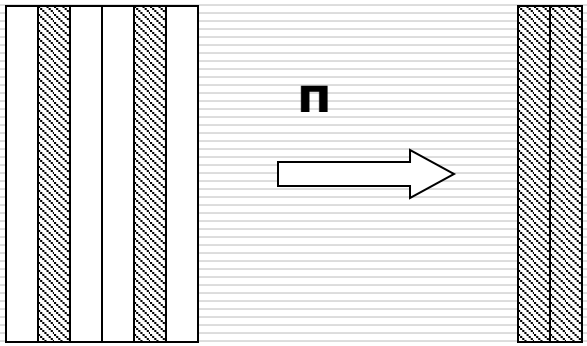
投影运算

- 一个一元运算，返回的还是作为参数的那个关系，但是忽略了某些属性。

- 符号：$\Pi_{A_1, A_2, A_3, \ldots, A_k}(r)$
  - 其中，r 是作为参数的关系，而 $A_1, A_2, \ldots, A_k$ 是 r 中的属性名

- 重复的行会被删除，因为关系是集合。

- 投影操作主要是从列的角度进行运算

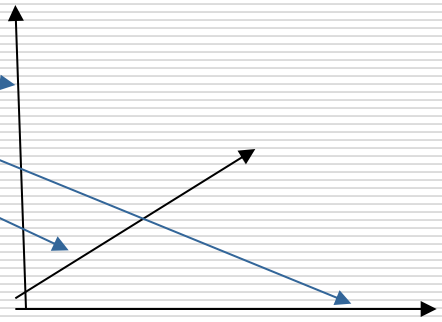


  - 但投影之后不仅取消了原关系中的某些列，而且还可能取消某些元组（避免重复行）

Project Operation – Example 投影

- Relation *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

去重复

=

A	C
α	1
β	1
β	2

6.1.3 关系运算的复合

- 关系代数运算的结果也是一个关系，所以它又可以用来做另一个关系代数运算的参数。
- 一般来说，可以将关系代数运算复合在一起，称为关系代数表达式。
- 例如：查找"Physics"系所有教师的姓名

$$\Pi_{name}(\sigma_{dept\_name="Physics"} (instructor))$$

- 小组讨论：
  - 利用University数据库中的表，设计一个复合运算？

6.1.4 笛卡尔积运算

- 小组讨论：
  - Cartesian-Product运算有什么特点？
  - Cartesian-Product运算有什么作用？
  - 为instructor和course表设计一个Cartesian-Product运算？

笛卡尔积运算

- 笛卡尔积运算允许我们结合来自任意两个关系的信息
- 符号：$r_1 \times r_2$
- 例如：instructor 关系 与 teaches 关系的笛卡尔积记作：instructor × teaches
- 任意一对元组都将构造出一个结果元组：其中，一个元组来自 instructor 关系，一个元组来自 teaches 关系
- 由于教师的 ID 在两个关系中都出现了，我们通过将关系名作为属性的前缀，来区分这些属性：
  - instructor.ID
  - teaches.ID

Cartesian Product: *instructor X teaches*

instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009

inst.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2009
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2010
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2009
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2010
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2010
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2009
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
12121	Wu	Finance	90000	10101	CS-101	1	Fall	2009
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2010
12121	Wu	Finance	90000	10101	CS-347	1	Fall	2009
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2010
12121	Wu	Finance	90000	15151	MU-199	1	Spring	2010
12121	Wu	Finance	90000	22222	PHY-101	1	Fall	2009
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

Cartesian-Product Operation – Example

- Relations *r*, *s*:

n行

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

m行

- *r* x *s*:

n*m行

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

完全搭配：

6.1.5 连接运算

- 笛卡尔积 instructor × teaches 把 instructor 关系的每一个元组与 teaches 关系的每一个元组都结合在了一起
  - 但是大多数的结果行都是无意义的
- 如果想要查找讲授过某门课的教师和该课程的信息，就需要这样写：

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

连接运算

- 连接运算可以将选择运算和笛卡尔积运算合并到一个运算中：
  - 考虑关系 r(R) and s(S)，令"θ"为 R∪S模式上的一个谓词
  - 连接运算 r $\bowtie_\theta$ s 定义如下：

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

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

  - 可以等价地写为：

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

6.1.6 Union Operation 并运算

- 小组讨论：
  - Union运算有什么特点？
  - 为instructor表设计一个Union运算？

并运算

- 并运算可以合并两个关系的所有元组到一个关系中
- 符号：r ∪ s
- 为了让运算 r ∪ s 合法
  - r, s 必须同元(具有相同的属性个数)
  - 属性域必须相容(例如：r 关系的第 2 列和 s 关系的第 2 列必须来自相同域)

- 如：查找 2017 年秋季学期或 2018 年春季学期开设的所有课程，或是两个学期都开设了的课程的集合

$$\Pi_{course\_id}(\sigma_{semester=\text{“Fall”} \wedge year=2017}(section)) \cup \Pi_{course\_id}(\sigma_{semester=\text{“Spring”} \wedge year=2018}(section))$$

属性集合相同才能合并

Union Operation – Example

- Relations *r, s:*

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- r ∪ s:

A	B
α	1
α	2
β	1
β	3

去重复

属性集合相同
才能合并

Set Difference Operation 差运算

- 小组讨论：
  - Set-Difference运算有什么特点？
  - 为instructor表设计一个Set-Difference运算？

集合差运算

- 找到在一个关系中，但不在另一个关系中的所有元组
- 符号：r – s
- 集合差也是要在相容关系之间进行运算的
  - r, s 同元
  - r 和 s 的对应属性域是相容的
- 例如：查找 2017 年秋季学期开设，但是 2018 年春季学期没有开设的课程的集合

$$\Pi_{course\_id}(\sigma_{semester="Fall" \wedge year=2017}(section)) - \Pi_{course\_id}(\sigma_{semester="Spring" \wedge year=2018}(section))$$

结果见图6-7.

Set difference of two relations

- Relations *r*, *s*:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- *r – s:*

A	B
α	1
β	1

Set-Intersection Operation 交运算

- 小组讨论：
  - 交运算有什么特点？

交运算

- 找到同时在两个输入关系中都出现的元组
- 符号：$r \cap s$
- 假定：
  - r, s 同元
  - r 和 s 的对应属性域是相容的
- 例如：查找 2017 年秋季学期和 2018 年春季学期都开设过的课程的集合

$$\Pi_{course\_id}(\sigma_{semester="Fall" \wedge year=2017}(section)) \cap \Pi_{course\_id}(\sigma_{semester="Spring" \wedge year=2018}(section))$$

结果见图6-6.

Set-Intersection Operation – Example

- Relation *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

6.1.7 赋值运算

- 有时通过将一个关系代数表达式中的一部分赋值给临时的关系变量，可以方便地编写该表达式
- 符号： $\leftarrow$
- 例如：查找"Physics"和"Music"系的所有教师

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

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

$$Physics \cup Music$$

- 使用赋值运算可以将一个查询编写成一个顺序程序，由一系列赋值后跟上一个表达式组成。

6.18 更名运算

- 关系代数表达式的结果并没有一个可以用来指代它们的名称。更名运算 ρ 可以给结果关系一个名字。
- 符号：$\rho_x(E)$，它返回表达式 E 的结果，并用 x 为结果关系命名。
- 另一种形式是：

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

它返回表达式 E 的结果，用 x 为结果关系命名，并用 $A_1, A_2, \ldots, A_n$ 给结果关系中的属性依次命名。

6.1.9 等价查询

- 用关系代数来编写查询的方式通常不止一种：
  - 例如：查找"Physics"系的教师所讲授课程的有关信息，一个查询表达式是：

    $\sigma_{dept\_name="Physics"}(instructor \bowtie_{instructor.ID=teaches.ID} teaches)$

  - 另一个查询表达式是：

$(\sigma_{dept\_name="Physics"}(instructor)) \bowtie_{instructor.ID=teaches.ID} teaches$

- 上面这两条查询不完全相同，但却是等价的 -- 也就是说，它们在任何数据库上都将给出相同的结果。

R_1

A	B	C
a	1	2
a	2	1
b	2	1
c	4	5

R_2

A	B	C
a	1	2
c	1	2
b	4	5
a	2	1

R_3

A	B
d	1
d	2

Exercise:

求$R_1 \cup R_2$, $R_1 - R_2$, $R_2 - R_1$, $R_1 \cap R_2$

$R_1 \times R_2$, $R_1 - R_3$, $R_1 \times R_3$

Any Question ?