Rational Algebra

六个基本运算符和四个复合操作

基本概念：

- Domain：属性的类型
- 没有重复项，relation是tuples的集合

SQL is declarative, relational algebra is procedural

一元操作符

Select

$\sigma_p(R)$ (sigma)

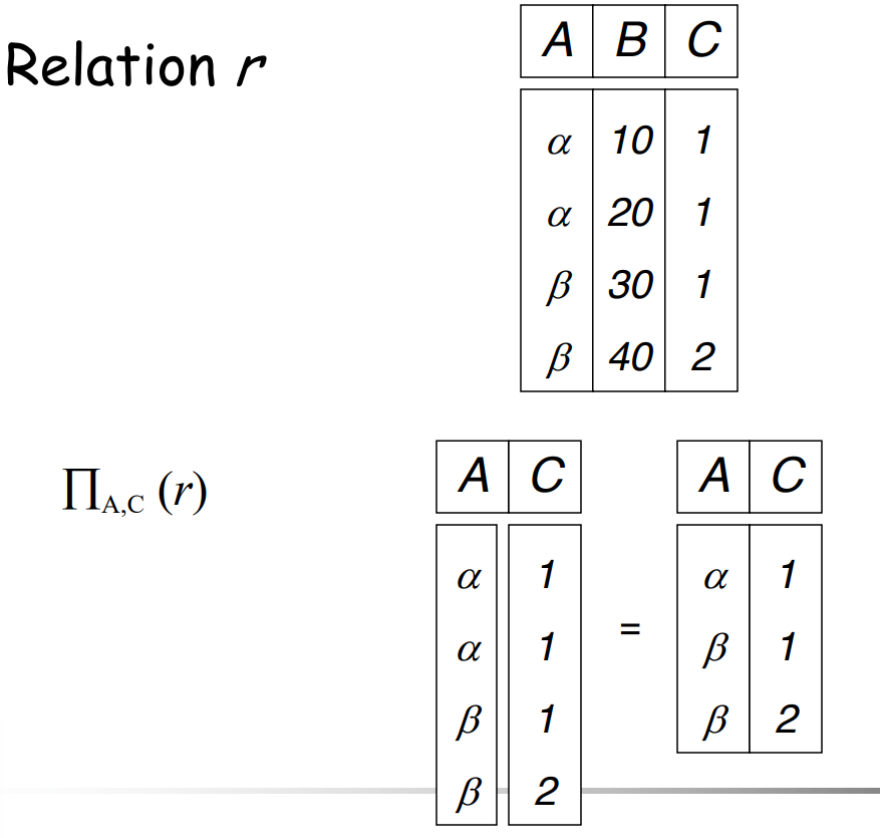
p is called the selection predicate

- Retains only a subset of **rows** (horizontal) 等效于WHERE子句
- Example: becomes `SELECT * FROM R WHERE id = 100` $\sigma_{id=100}(R)$

Projection

π (pi)

- Retains only desired **columns** (vertical) and **erase** the columns that are not listed等效于SELECT子句
- Example: becomes `SELECT name FROM R` $\pi_{name}(R)$



Rename

ρ

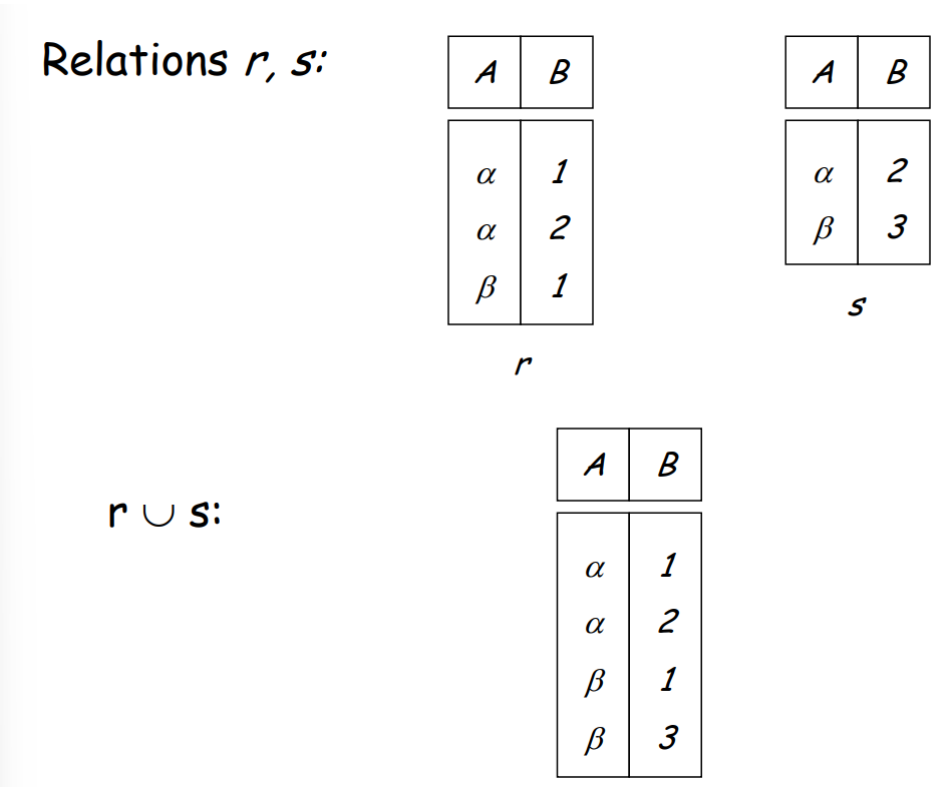
- rename attributes and relations 方便引用以保持语义清晰
- Example: $\rho_{(1\to sid1,4\to sid2)}(R)$ renames the 1st and 4th columns to and respectively `sid1` `sid2`

二元操作符

Union

$\cup$

- Or operator: either in r1 or r2
- Equivalent to in SQL (doesn't keep duplicates: does) `UNION` `UNION ALL`
- 条件：
  1. 属性数量相同
  2. 属性domain相同



Set difference

—

- Tuples in r1, but not in r2
- Equivalent to in SQL `EXCEPT`
- 条件同Union

Relations r, s

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

$r - s$

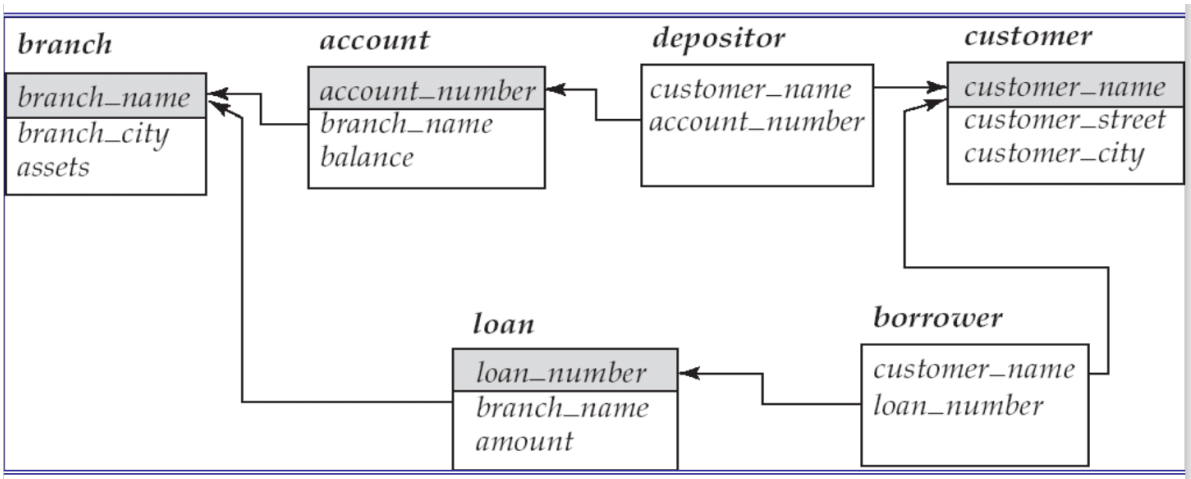
A	B
α	1
β	1

Cartesian-Product

×

- Joins r1 with all r2
- Equivalent to in SQL `FROM r1, r2...`
- 假设两个表没有相同属性，否则要rename

example

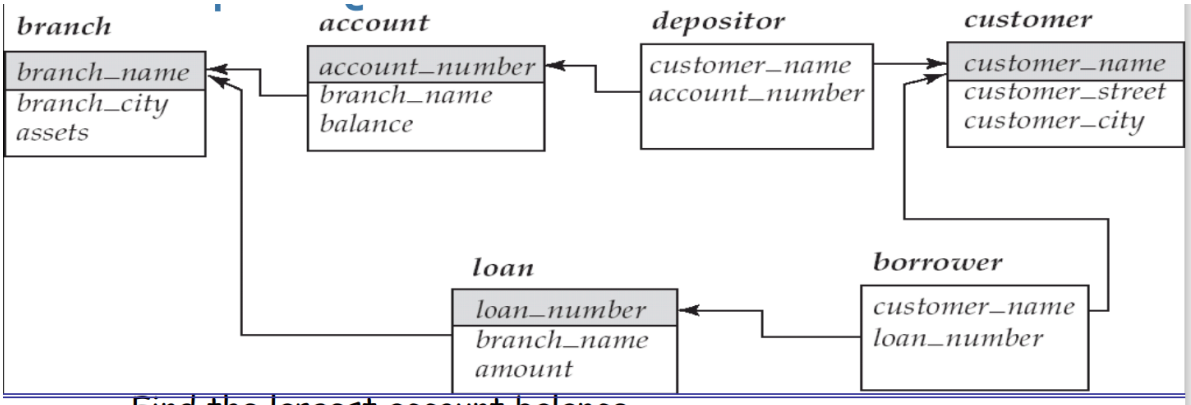


Find the names of all customers who have a loan at the Perryridge branch.

$$\Pi_{customer\_name}(\sigma_{branch\_name = \text{“Perryridge”}}(\sigma_{borrower.loan\_number = loan.loan\_number}(borrower \times loan)))$$

OR

$$\Pi_{customer\_name}(\sigma_{loan.loan\_number = borrower.loan\_number}((\sigma_{branch\_name = \text{“Perryridge”}}(loan)) \times borrower))$$



- Find the largest account balance
  - Strategy:
    - Find those balances that are *not* the largest
      - Rename *account* relation as *d* so that we can compare each account balance with all others
    - Use set difference to find those account balances that were *not* found in the earlier step.

$$\Pi_{balance}(account) - \Pi_{account.balance}(\sigma_{account.balance < d.balance}(account \times \rho_{d}(account)))$$

additional operations

用以简化查询的复合操作（宏）

Intersection

$\cap$

- And operator: both in r1 and r2
- 条件同union

Natural Join

$\bowtie$

属性是两个关系的union，tuples是两个关系的笛卡尔积中匹配属性相同的行

Combine relations that satisfy predicates

equi-join on all matching column name

$R \bowtie S = \pi_{uniquecols}\sigma_{matchingcolsequal}(R \times S)$

Relations 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	$\in$

s

r ⋈ 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

DIvision

÷

属性是两个关系的difference，tuples在R里找S

- in SQl `for all`

Relations *r, s*

r

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

s

D	E
a	1
b	1

r ÷ s

A	B	C
α	a	γ
γ	a	γ

Assignment

一种**语法工具**，用于表达式的简化和管理。