Cartesian-Product operation

- Combine information, r_1 x r_2
- Remember: a relation is a subset of a Cartesian product
- Naming scheme to differentiate attributes: `relation.attribute`
    - only for non-distinct attributes
- What tuples appear in r_1 x r_2 ?
- tuples in r_1 x r_2 : all possible combinations of tuples in r_1 and r_2
- if r_1 has n_1, and r_2 has n_2, then r_1 x r_2 has $n_1 * n_2$ tuples

Cartesian-Product

- For relations $r_1(R_1)$, $r_2(R_2)$:
  - $r_1 \times r_2$ concatenation of R_1 and R_2
- For tuple $t \in r_1 \times r_2$,, then:
  - there is $t_1 \in r_1$ and $t_2 \in r_2$ such that:
  - $t[R_1] = t_1[R_1]$ and $t[R_2] = t_2[R_2]$

Rename Operation

- Name and refer to results of relational-algebra operations
- Allows us to refer to a relation by more than one name.
- Example:

$$\rho_x(E)$$

returns the expression E under the name X

- If a relational-algebra expression E has arity n, then

$$\rho_{x(A1,\ A2,\ \ldots,\ An)}(E)$$

returns the result of expression E under the name X, and with the attributes renamed to A_1, A_2 , …., A_n .

Example of Rename in action

Query: Find the highest amount

$$loan \times \rho_{temp}(loan)$$

Cartesian product with itself (renamed)

↓

$$\sigma_{loan.amount < temp.amount}(loan \times \rho_{temp}(loan))$$

Rows without the highest amount

↓

$$\pi_{loan.amount}(\sigma_{loan.amount < temp.amount}(loan \times \rho_{temp}(loan)))$$

↓

$$\pi_{balance}(account) - \pi_{loan.amount}(\sigma_{loan.amount < temp.amount}(loan \times \rho_{temp}(loan)))$$

loan_number	amount
L-11	900
L-15	1500
L-17	1000
L-93	500

Loan table

Another rename example

Customers living in the same street and city as Green

Customer_name	Customer_street	Customer_city
Adams	Spring	Chicago
Brooks	Senator	Brooklyn
Curry	Elm	Harrison
Glenn	New Era	Stamford
Green	Elm	Harrison
Hayes	Elm	Harrison

Find out Green's street and city:

$\pi_{customer\_street,\ customer\_city}(\sigma_{customer\_name=\text{"Green"}})$ ..... A

Rename it: (why?)

$\rho_{green\_addr(street,\ city)}((A))$ ..... B

Take cross-product with original relation

Customer X B C

Select needed values and project

$\pi_{customer.customer\_name}(\sigma_{customer.customer\_street=green\_add.street\ \wedge\ customer.customer\_city=green\_add.city}(C))$

Additional (derived) operations

- Set-Intersection operation

  Customers who have both a loan account and a borrower account

  $$\Pi_{customer\_name}(borrower) \cap \Pi_{customer\_name}(depositor)$$

- $r \cap s = r - (r - s)$

customer_name
Hayes
Johnson
Jones
Lindsay
Smith
Turner

Depositor

$\cap$

customer_name
Adams
Curry
Hayes
Jackson
Jones
Smith
Williams

Borrower

→

customer_name
Hayes
Jones
Smith