Lecture-32
DC

classmate
Date 12/04/18
Page

Relational algebra

Fundamental ~~algebra~~ operators

1. Select (σ) $\sigma_{F}(r)$
2. Project (π)
3. Cartesian product (x)
4. Rename (ρ)
5. Union (∪)
6. Set difference (−)

Derived

1. Intersection (∩)
2. Join (⟕, ⟖, ⋈, ⟗)
3. Division (÷)

Select

This oper^n returns a set of tuple satisfying a given predicate.

Ex- Return names of those teachers who teach database.

$$\underset{\text{cond}^n}{\underset{\downarrow}{\sigma_{\text{course}='\text{DBMS}'}}}\ \underset{\searrow\text{relation}}{(\text{teach})}$$

Project

~~Selecting names~~

This oper^n allows display of any no. of columns from a given relation

Ex - select names

$$\underset{\text{attrib to select}}{\underset{\downarrow}{\pi_{\text{tname}}}}\left(\sigma_{\text{course}='\text{DBMS}'}(\text{teach})\right)$$

→ select & project interchange meanings in SQL & relational algebra.

$\pi_{1,2,3}(S)$ → 1, 2, 3rd column

Rename (ρ)

$$\underset{\text{new name}}{\underset{\downarrow}{\rho_{x}}}(\text{student})$$

Cartesian product (x) → cross join equivalent

Set difference → No. of arity must be same.
i.e., no. of attrib are same

R	S	R ∪ S	R − S
a b c	b c a	a b c	a b c
d c a	c d f	d c a	d c a
b f e	b f e	b f e	
		b c a	
		c d f	

Relation algebra queries
branch (branch-name, branch-city, assets)
customers (cust-name, cust-street, cust-city)
account (acc-no, branch name, balance)
loan (loan-no, branch-name, amount)
depositor (cust-name, acc-no)
Borrower (cust-name, loan-no)

i) Find all loan above Rs 12000.
ii) Find loan no of each loan with amount greater than 12000.
iii) Find names of all customers who have a loan, or an account or both from a bank
iv) Find names of all customers who have a loan at the Park street Branch
v) Find names of all customers " " " "
" " " but do not have account at any branch of the bank.
vi) Find the largest acc. balance.
vii) Find all customers who have an account at all branches located in Bangalore.

i) $\sigma_{amt > 12000}(loan)$
ii) $\pi_{loan\text{-}no}(\sigma_{amt > 12000}(loan))$
iii) $\pi_{cust\text{-}nam}(borrower) \cup \pi_{cname}(depositor)$
iv) $\pi_{cust\text{-}name}(\sigma_{branch\text{-}name = 'Park\ street'}(borrower \bowtie loan))$

v) $\pi_{cname}(\sigma_{branch\_name = 'Park\ street'}(borrower \bowtie loan)) - \pi_{cname}(depositor)$

vi) Steps

i) Find those balances that are not the largest.

ii) Rename accnt. relation as 'd' so that we can compare each accnt. balance with others.

iii) Use set diff to find those accnt. balances that were not found in earlier steps.

iv) $\pi_{balance}(account) - \pi_{account.balance}(\sigma_{account.balance < d.balance}(account \times \rho_{d}(account)));$

$$\pi_{cust\_name,\ branchname}(depositor \bowtie account) \div \pi_{branch\_name}(\sigma_{branch\_city = 'Bangalore'}(branch))$$

R

A	B	C	D
b	c	e	f
a	b	i	j
b	c	g	h
b	c	a	d
d	i	g	h
d	i	j	k
d	i	e	f

S

C	D
e	f

$R \div S = \pi_{1,2}(R) - \pi_{1,2}(\pi_{1,2}(R) \times S - R)$

Steps

i) $\pi_{1,2}(R)$

ii) $\pi_{1,2}(R) \times S$

iii) $\pi_{1,2}(R) \times S - R$

iv) $\pi_{1,2}$ of step 3 result

v) $\pi_{1,2}(R)$ - result of step 4