Database Management System (DBMS)

Lecture-17

Dharmendra Kumar

September 10, 2020

Intersection Operation

Example: Find all customers who have both a loan and an account.

Solution: $\Pi_{customer}(depositer) \cap \Pi_{customer}(borrower)$

customer-name
Hayes
Jones
Smith

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

Join

Join is a combination of a Cartesian product followed by a selection process. A Join operation pairs two tuples from different relations, if and only if a given join condition is satisfied.

Types of join operations

- Theta (θ) Join or Condition Join
- Equijoin
- Natural-Join

Theta (θ) Join or Condition Join

Theta join combines tuples from different relations provided they satisfy the theta condition. The join condition is denoted by the symbol θ.

$R_1 \bowtie_\theta R_2$ R_1 and R_2 are relations having attributes $(A_1, A_2, .., A_n)$ and $(B_1, B_2, .., B_n)$ such that the attributes don't have anything in common, that is $R_1 \cap R_2 = \phi$.

Theta join can use all kinds of comparison operators.

Equijoin

When Theta join uses only equality comparison operator, it is said to be equijoin. The above example corresponds to equijoin.

Example: Theta join and Equijoin operations are shown as following:-

(*sid*)	*sname*	*rating*	*age*	(*sid*)	*bid*	*day*
22	Dustin	7	45.0	58	103	11/12/96
31	Lubber	8	55.5	58	103	11/12/96

$$S1 \bowtie_{S1.sid<R1.sid} R1$$

sid	*sname*	*rating*	*age*	*bid*	*day*
22	Dustin	7	45.0	101	10/10/96
58	Rusty	10	35.0	103	11/12/96

$$S1 \bowtie_{R.sid=S.sid} R1$$

Natural-Join

- Natural join does not use any comparison operator. It does not concatenate the way a Cartesian product does. We can perform a Natural Join only if there is at least one common attribute that exists between two relations. In addition, the attributes must have the same name and domain.
- Natural join acts on those matching attributes where the values of attributes in both the relations are same.

Example: Natural join operation is shown as following:-

- 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 $\bowtie$ 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

Example: Find the names of all customers who have a loan at the bank, along with the loan number and the loan amount.

Solution: Query without using natural join is

$\Pi_{customer-name, loan-number, amount}(\sigma_{borrower.loan-number=loan.loan-number}(bor$
$loan))$

Equialent query using natural join is

$\Pi_{customer-name, loan-number, amount}(borrower \bowtie loan)$

customer-name	*loan-number*	*amount*
Adams	L-16	1300
Curry	L-93	500
Hayes	L-15	1500
Jackson	L-14	1500
Jones	L-17	1000
Smith	L-23	2000
Smith	L-11	900
Williams	L-17	1000

Relational Algebra

Example: Find the names of all branches with customers who have an account in the bank and who live in Harrison.

Solution: $\Pi_{branch-name}(\sigma_{customer-city=\text{''}Harrison\text{''}}(customer \bowtie depositor \bowtie account))$

branch-name
Brighton
Perryridge

Note: If there is no common attributes between two relations, then natural-join and Cartesian product is equal.

The Division Operation

- The division operation, denoted by , is suited to queries that include the phrase "for all."
- We are describing division operation through an example. Consider two relation instances A and B in which A has (exactly) two fields x and y and B has just one field y, with the same domain as in A. We define the division operation A/B as the set of all x values (in the form of unary tuples) such that for every y value in (a tuple of) B, there is a tuple $< x, y >$ in A.
- Another way to understand division is as follows. For each x value in (the first column of) A, consider the set of y values that appear in (the second field of) tuples of A with that x value. If this set contains (all y values in) B, then the x value is in the result of A/B.

Example: Division operation is explain in the following figure:-

A

sno	*pno*
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

B1

pno
p2

B2

pno
p2
p4

B3

pno
p1
p2
p4

A/B1

sno
s1
s2
s3
s4

A/B2

sno
s1
s4

A/B3

sno
s1

Relational Algebra

Example: Find all customers who have an account at all the branches located in Brooklyn.

Solution: In this query, we will apply the division operator. For this we have to find numerator and denominator of the query. If numerator is N and denominator is D then final query will be N$\div$ D.

In this query, denominator is all the branches located in "Brooklyn". Query for this is

$$D = \Pi_{branch-name}(\sigma_{branch-city="Brooklyn"}(branch))$$

branch-name
Brighton
Downtown

Result of $\Pi_{branch\text{-}name}(\sigma_{branch\text{-}city = \text{“Brooklyn”}}(branch)$.

And numerator is all the customers who have an account with their branch name. Query for this is

N $= \Pi_{customer-name, branch-name}(depositor \bowtie account)$

customer-name	*branch-name*
Hayes	Perryridge
Johnson	Downtown
Johnson	Brighton
Jones	Brighton
Lindsay	Redwood
Smith	Mianus
Turner	Round Hill

Result of $\Pi_{customer\text{-}name,\ branch\text{-}name}\ (depositor \bowtie\ account)$

Therefore, the final query is

N ÷ D.

Customer-name
Johnson

Result of final query

Note: Let r(R) and s(S) be given, with S ⊆ R:

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

Assignment Operation

It is denoted by ←. If E is a relational algebra query expression, then we can assigned it as like the following:-

r ← E